IRNAS Rače – Institute for Development
of Advanced Applied Systems Rače, Slovenia
**email**: info@irnas.eu, **web**: www.irnas.eu

# Data Fit to 2-D Gaussian Function

9. april 2015

## 1 Introduction

Often we encounter data of the known form, resembling the Gaussian distribution. Having a sufficient sample of measurements, in the case of one-dimensional data, most of computational software allows us to find quite accurate fit to the representative Gaussian function. However, in the case of more dimensional data, there are very few pre-build functions available to help us determine accurate enough approximation. To make fitting even harder, assume only limited measurement sample of random points (not a whole mesh) is known.

Matlab Curve Fitting Toolbox does include a 'fittype' function, which enables curve and surface fitting to a custom or predefined model. In particular, it allows fitting a 2-D data to a Gaussian function. User can also customise settings, such as initial guess, limits of solution and more. But if one is limited to use an open-source software, such as Octave, there are currently no similar options available.

In this short article we present a possible approach for a specific case, where only a small measurement sample is known. A fit to a Standard 2-D Gaussian distribution of the form

$$f(x,y) = A\exp\left(-\frac{(x-x_0)^2}{2\delta_x^2} - \frac{(y-y_0)^2}{2\delta_y^2}\right), \tag{1}$$

where

- $A$: the amplitude (height) of distribution
- $(x_0, y_0)$: centre of distribution
- $(\delta_x, \delta_y)$: $x$ and $y$ standard deviations, i.e. spreads

is constructed. Based on the size of measurement sample and distribution on measurement points, we should not expect a very accurate fit, more an approximate model that can provide us with more information about the data.

### 1.1 The Problem

In the particular case, our data sample consists of laser beam power measurements on 9 pre-defined points, located on the 10x10mm surface, were laser is pointing. We know the beam has Gaussian distribution and that its centre should lie somewhere on the measurement surface.

The aim is to find an approximate 2-D Gaussian function, that would allow us to determine x and y diameters of the beam. An assumption that $\delta_x = \delta_y$ is made, based on the laser properties in order to simplify the problem. The constructed solution is not limited to Octave and can be implemented in any computational software and most programming languages, as it does not contain any build-in functions, that can not be easily written in other programming languages.

## 2 Mathematical Derivation

We begin looking at the Gaussian of the form

$$f(x,y) = A\exp\left(-\frac{(x-x_0)^2}{2\delta^2} - \frac{(y-y_0)^2}{2\delta^2}\right), \tag{2}$$

taking natural logarithm of both sides

$$\ln(f(x,y)) = -\frac{(x-x_0)^2}{2\delta^2} - \frac{(y-y_0)^2}{2\delta^2} + \ln(A). \tag{3}$$

Equation (3) can be rewritten in the form

$$\ln(f(x,y)) = K_1(x^2+y^2) + K_2x + K_3y + K_4, \tag{4}$$

where

$$K_1 = -\frac{1}{2\delta^2} \tag{5}$$

$$K_2 = \frac{x_0}{\delta^2} \tag{6}$$

$$K_3 = \frac{y_0}{\delta^2} \tag{7}$$

$$K_4 = -\frac{x_0^2+y_0^2}{2\delta^2} + \ln(A). \tag{8}$$

The above coefficients can be determined solving system of linear equation. In particular, if $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ are column vectors containing measurements, we define a matrix and a vector

$$A = [\mathbf{x}^2 + \mathbf{x}^2, \mathbf{x}, \mathbf{y}, \mathbf{1}] \qquad \mathbf{b} = \ln(\mathbf{z}). \tag{9}$$

Then the coefficients are solution to the matrix equation

$$A \cdot \mathbf{K} = \mathbf{b}. \tag{10}$$

The above equation defines a system of linear equations, which can be solved in many different ways. In Octave we we can easily obtain $\mathbf{K}$ by simply calculating $A \setminus \mathbf{b}$ or using function 'linsolve'. Note that more advanced methods should be used in order to obtain more accurate solution, specially as the system is overrefined in the most cases.

Once coefficients are calculated, we need to reverse them back to parameters defining Gaussian function. We have

$$\delta = \sqrt{-\frac{1}{2K_1}} \tag{11}$$

$$x_0 = -\frac{K_2}{2K_1} \tag{12}$$

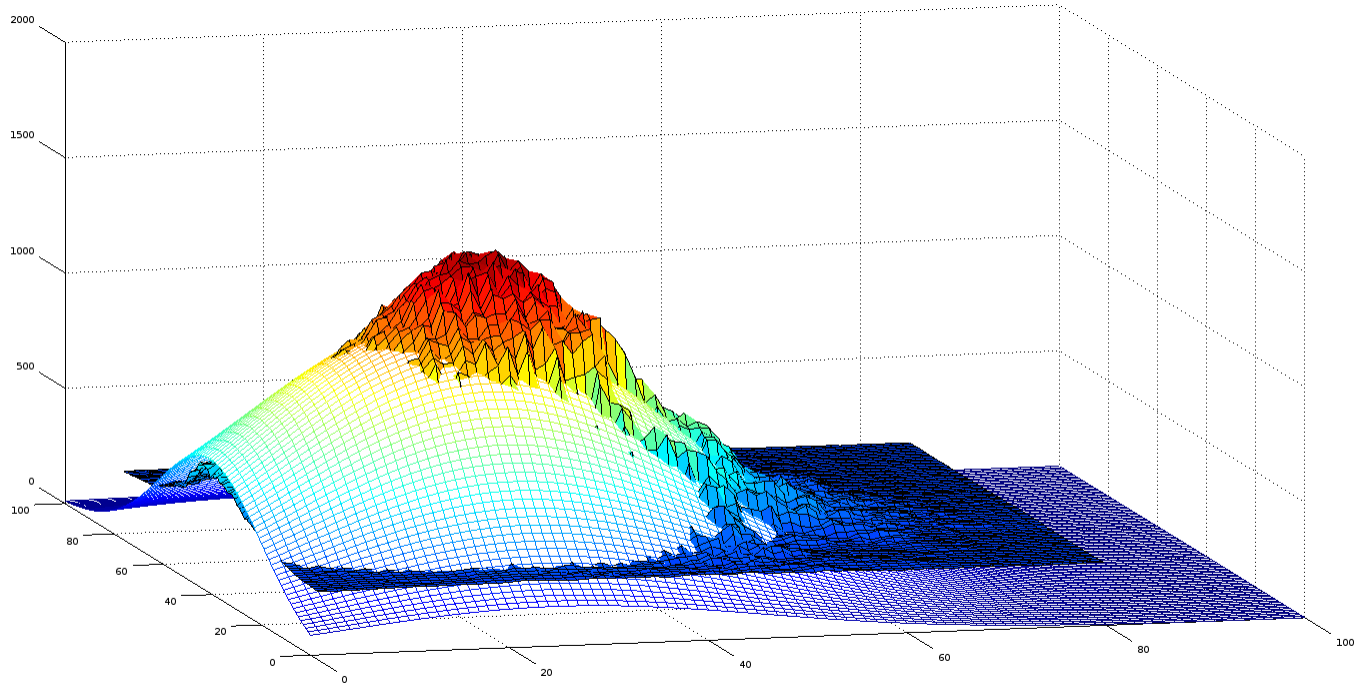
$$y_0 = -\frac{K_3}{2K_1} \tag{13}$$

$$A = \exp\left(K_4 + \frac{K_2^2 + K_3^2}{8K_1^3}\right). \tag{14}$$

## 3 Uses, Accuracy and Possible Improvements

Using this method, one should not expect very accurate and reliable results, as often only a small set of data measurement points and noisy data are available.

Gaussian fit to the power measurement of the Gaussian laser beam is displayed below (Figure 1). Fit was made based on 9 measurement points, one in the centre, other 8 in two concentric circles, located on the 100x100 mm grid.




Slika 1: Comparison: Beam power measurement and Gaussian fit

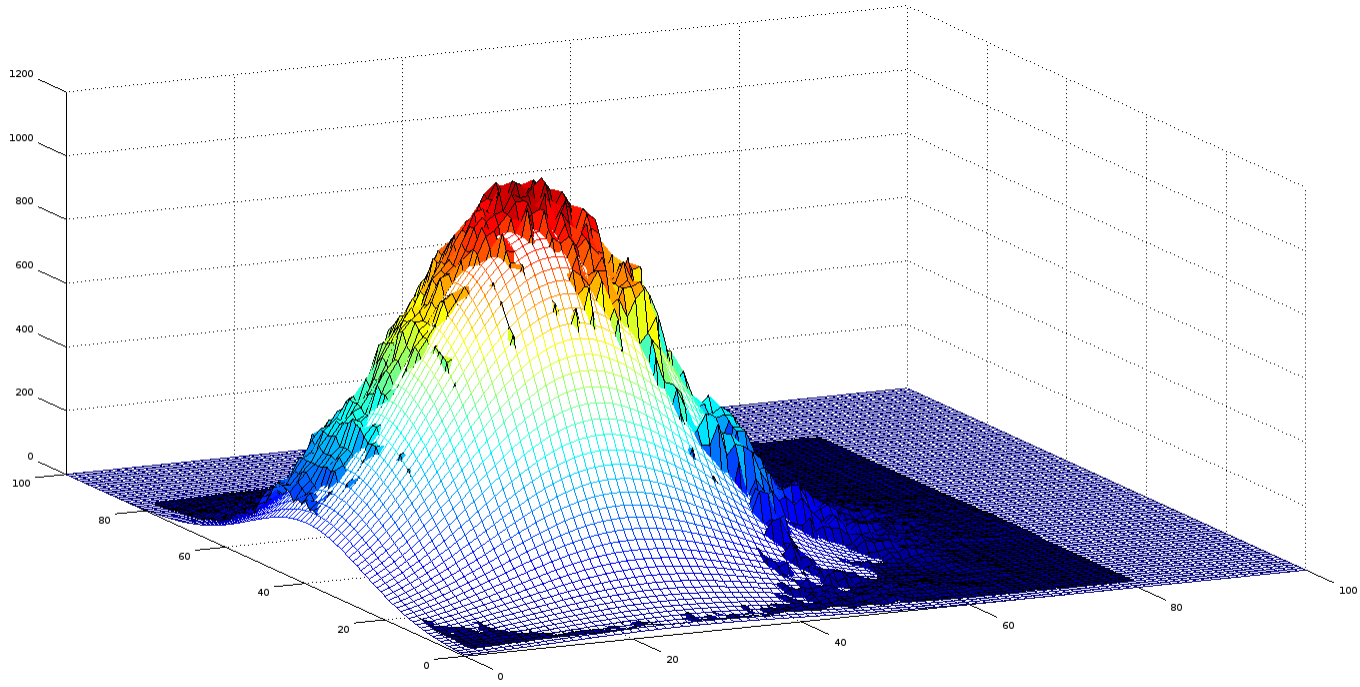
As we can see fit is not very accurate, however still useful for approximately locating the centre

of the beam and determining its diameter (set appropriate threshold).

One of the problems is that our data is elevated and never reaches the plane $z = 0$. Having more knowledge about data structure, we can improve fit by shifting the data for appropriate value (if known) and run the algorithm. Another, more general improvement, that can be made is to set bounds on the calculated coefficients. For example

- $A$: we can find maximum measurement in our data set and use it as a lower bound for the calculated amplitude
- $(x_0, y_0)$: if we approximately know where the centre of distribution should be located, bounds can be set for $x_0$ and $y_0$ as well.

Implementing the above mentioned improvements, we obtain a better fit for our Gaussian beam as seen in the Figure 2




Slika 2: Improved fit: Beam power measurement and Gaussian fit

### 3.1 Octave Implementation

A code, implementing a basic algorithm described above, is available on GitHub. The function 'gaussian2d' excepts data in the form of $3xn$ matrix as an input argument, and returns array with coefficients.

Further improvements and modifications based on individual data are encouraged. One must also note, that very small set of measurement points, located closely, not distributed evenly and thus representing just small part of data, will not produce any relevant results.